

iMesh CCA Component

The Interoperable Technologies For Advanced Petascale Simulations (ITAPS) Center has defined a common mesh interface, **iMesh**, for managing a discrete mesh composed of sets of entities, such as nodes, edges, faces and volumes, modifications to these entities and tags and tag data associated sets of entities. Detailed information is available at [the ITAPS website](#).

A CCA-compliant iMesh component has been written for this project. The component is a direct mapping to the iMesh C language interface defined by ITAPS and the underlying library is MOAB. Tests and optimizations are ongoing.

Building the Component

Building the component should be as simple as creating a CCA port and using the iMeshC.iMeshPort.sidl file attached to this article. After a port is created, a component that provides this port can be implemented with the attached iMeshC_iMeshComponent_Impl.c file.

Design and Performance

The SIDL file for this component was developed from a previous SIDL wrapping included with MOAB. The most important difference is that this version does not use objects and, instead, passes an opaque pointer to the mesh with each function call. The component is implemented in C and is a direct 1-1 wrapping of the iMesh C interface.

There are some important memory considerations when wrapping a library with SIDL. In particular, SIDL defines its own array type and the efficient transfer of information to and from this type must be considered to avoid a performance hit. In this component, the following "tricks" are used:

- * Copying incoming arrays can be avoided if the stride of the SIDL array is equal to one. If this is true, then it is sufficient to pass the address of the first element.
- * Outgoing arrays can always be "borrowed," so long as the memory doesn't magically disappear.

Tests

The iMesh CCA component has not been extensively tested in applications, but it has passed the tests in a component implementation of the TSTT test program. The component has only been tested with the MOAB library.

A few changes were made to the TSTT test program during the componentization. Most of these changes were made to test extra functions that were not originally tested in the TSTT program. A few changes were made to appease SIDL. All tag names, values, etc. were not changed.

A very small performance test is presented below. This is a test of the amount of time needed to return all of the vertex coordinates. This is currently tested in two ways:

- * Use the "1 step" call to getAllVtxCoords in iMesh and its analogs in the component and MOAB
- * Use a "2 step" method where first the entity handles are obtained by looking for objects of type "vertex" with getEntities and the coordinates are then gathered by getVtxArrCoords.

See the graph below. The "2 step" method is faster, mostly because the "1 step" method in iMesh requires that an "in_entity_set" array be filled by iterating across all the vertices.

Contact

- Jay Billings, ORNL